ON SOME COMMUNICATION NETWORK PROBLEMS

Robert Kalaba
Engineering Division
The RAND Corporation

P-1325

22 April 1958
Revised 3 June 1959

Reproduced by

The RAND Corporation • Santa Monica • California

The views expressed in this paper are not necessarily those of the Corporation

CLEARINGHOUSE FOR FEDERAL SCIENTIFIC ANP TECHNICAL INFORMATION CFSTI
DOCUMENT MANAGEMENT BRANCH 410.11

LIMITATIONS IN REPRODUCTION QUALITY

AD 606 555

ACCESSION #

☒ 1. WE REGRET THAT LEGIBILITY OF THIS DOCUMENT IS IN PART UNSATISFACTORY. REPRODUCTION HAS BEEN MADE FROM BEST AVAILABLE COPY.

□ 2. A PORTION OF THE ORIGINAL DOCUMENT CONTAINS FINE DETAIL WHICH MAY MAKE READING OF PHOTOCOPY DIFFICULT.

□ 3. THE ORIGINAL DOCUMENT CONTAINS COLOR, BUT DISTRIBUTION COPIES ARE AVAILABLE IN BLACK-AND-WHITE REPRODUCTION ONLY.

□ 4. THE INITIAL DISTRIBUTION COPIES CONTAIN COLOR WHICH WILL BE SHOWN IN BLACK-AND-WHITE WHEN IT IS NECESSARY TO REPRINT.

□ 5. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED, DOCUMENT WILL BE AVAILABLE IN MICROFICHE ONLY.

□ 6. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED DOCUMENT WILL NOT BE AVAILABLE.

□ 7. DOCUMENT IS AVAILABLE IN MICROFICHE ONLY.

□ 8. DOCUMENT AVAILABLE ON LOAN FROM CFSTI (TT DOCUMENTS ONLY).

□ 9.

NBS 9 64

PROCESSOR:

SUMMARY

The general field of communication provides a rich source of combinatorial problems, a number of which arise in connection with the design and utilization of communication networks. Several classes of extremal problems are discussed including the leasing of minimal cost spanning networks, the finding of optimal paths through networks, and the optimal routing of messages in networks, along with various generalizations.

The methods employed involve curious admixtures of the functional equation approach of dynamic programming, linear programming, and various *ad hoc* procedures. Interest centers on obtaining methods which are either efficient from the point of view of machine computation or which emphasize the underlying structure of the solutions.

I. INTRODUCTION

The general field of communication provides a rich source of problems in applied mathematics. These embrace fundamental considerations of the communication process itself, [30], a wide spectrum of scientific and technological problems, and still others involving the design and utilization of large-scale networks. The rather modest objective of this paper is to draw attention to several classes of communication network problems, of some importance in the applications, which lead to combinatorial problems of varying degrees of complexity. Generally speaking, these problems are concerned with the optimal design and utilization of communication networks in which the complex interactions among users' demands for service, system capacities, and economic factors must be resolved. Pioneering efforts along these lines are associated with the names of A. K. Erlang, [7], T. C. Fry, [18], E. C. Molina, [27], R. I. Wilkinson, [33], and J. Riordan, among others.

Problems of the type mentioned have been assuming increasing importance in recent years due to the rapid expansion of communication systems, involving large capital investments. Wide-sweeping technological improvements in both switching and transmission facilities will vastly alter the nature of the networks. Lastly, the advent of the high-speed large-memory digital computing machine has forced a re-evaluation of the very methods of analysis and design which are in current use. Though the models which we shall discuss are highly simplified, their analysis may point the way toward the treatment of more refined and realistic ones.

The problems to be discussed lead, from the mathematical point of view, to the determination of extrema from among finitely many choices, so that no questions concerning existence of solutions arise. Interest centers rather on obtaining algorithms which lead to efficient computational schemes for obtaining solutions, and which shed light on the structure of the solutions. Finding solutions in these problems through the mere enumeration of cases, as remarked by Euler in his famous paper on the Königsberg bridge problem, is at best onerous and unsatisfying and in many situations impossible (even with the aid of a high-speed computing machine), as will become evident.

The first type of problem which we shall consider is that of determining minimal cost connecting networks. Given a network, each link of which has a cost assigned to it, find a connected network which includes all the stations and has least total cost. Solutions have been proposed by Kruskal, [24], Prim, [29], and Kalaba in the forms of algorithms which lend themselves well to hand and machine computation and which provide much insight into the nature of the solution. This problem can be generalized along various lines.

The second type of problem is that of determining an optimal chain connecting two points in a network. Perhaps the simplest version of this type is to find a shortest chain connecting two terminals in a given network, each link of which has a prescribed time of transit. Solutions have been provided by Bellman, [2], through the use of functional equations, Dantzig, [10], using a linear programming approach, and Ford, [14]. The problem may be modified by requiring that the chain pass through several specified intermediate points, and it still remains amenable to treatment. Furthermore, Bellman and Kalaba have proposed a method for finding the n^{th} shortest chain leading from one point to another in a network, [5].

The methods to be discussed make possible the solution of certain optimal chain problems involving probabilitic considerations. In particular, the problem of determining a path through a network which maximizes the probability that the time of transit between two given points be no greater than a prescribed time t is solved, using functional equations, under the assumption that the times of traverse of the various branches are independent random variables with known probability densities. In addition some applications to the theory of blocking in networks are provided, [25].

The last type of problem discussed involves the optimal routing of messages in networks, [20]. Under certain conditions one may formulate this as a linear programming problem for which Dantzig's simplex method is available for numerical solution, provided the network is not too large. A method of solution based on an idea of Ford and Fulkerson, [16], makes possible the numerical solution of problems involving about 150 links. Finally, some related problems involving interoffice trunking and the augmentation of networks to meet increased demands for service are discussed, [21].

Acknowledgement. During the preparation of this paper the author has had the benefit of many discussions with R. Bellman, G. Dantzig, D. R. Fulkerson, and M. L. Juncosa and expresses his sincere thanks for their friendly interest and suggestions.

II. MINIMAL COST CONNECTING NETWORKS

1. Formulation.

A television broadcasting company wishes to lease video links so that its stations in various cities may be formed into a connected network. Assuming that the costs for the individual links, all different, are known, we wish to show how to construct the network at minimal cost, [1]. (Continuity considerations enable one to remove the restriction that the costs be different, but, as will become evident, uniqueness of the solution may be lost.)

Various solutions for this problem will now be discussed and some extensions will be indicated.

2. Solution I.

Kruskal, [24], has proposed the following solution, the simplicity of which is quite remarkable. Perform the following step as often as possible: Among the links not yet included in the connecting network, choose the lowest priced link which does not form any loops with the links already chosen. The proof, which follows, is by contradiction.

If there are N stations in the network, it is evident that a minimal cost connecting network, denoted by K, contains no loops and consists of exactly N-1 links. Let the links chosen according to the above algortithms be denoted by $e_1, e_2, \ldots, e_{N-1}$; since the costs are all different from each other, this sequence is uniquely determined. This set of links is denoted by L_{N-1}.

If $K \neq L_{N-1}$, let e_i be the link of lowest index of L_{N-1} which is not in K. If e_i is added to the set K, a loop is formed of which e_i is one link. This loop also contains a link, f, which is not in L_{N-1} but which is in K. Furthermore, the link f does not close a loop when added to the set e_1, e_2 ..., e_{i-1}, for all these links, including f, lie in the set K, which contains no loops. But according to the algorithm e_i is the lowest priced such link; consequently

(1) $\qquad$ price $(f) >$ price (e_i).

This implies that the network consisting of the union of K and e_i from which f has been deleted, which also contains N-1 links and does not contain any loops, is available at lower cost than K, contrary to assumption. Hence the Kruskal tree $L_{N-1} = K$ is the unique minimal cost connecting network.

3. Solutions II and III.

In the same paper referred to above, Kruskal also proposes two additional constructions. Let S be an arbitrary, but fixed and non-empty subset of all the N stations to be joined into a connected network. Perform the following step as often as possible: Among the links not yet chosen, but which are connected either to a station in S or to a link already chosen, choose the link of lowest price which does not form any loops with the links already chosen. This reduces to the construction of Section 2 if S consists of all the stations in the network.

The other consists in determining the links not in K by choosing as many times as possible, from among the links not yet chosen, the most expensive link which does not disconnect the network. The set of links not eventually chosen forms the minimal cost connecting network K. This may be established by showing that it is always possible to remove a link from consideration for membership in K if the link is the most costly link whose removal from the network does not disconnect it. Let A be the set of links which can be removed without disconnecting the network, and let e be the one of greatest cost. Suppose e to be in K. The removal of the link e from the set K disconnects this network, which can, however, be reconnected by the addition of a link f which is contained in the set A and is different from e; for if this were not the case, e could not be an element of the set A. Consequently, the union of K and f, from which e is deleted, would be available at lower cost than K, which results in a contradiction.

4. Solution IV.

Still another algorithm is available in which we proceed from one connecting network containing no loops to another of lesser cost until the optimal network K is attained. Select any connecting network with precisely N-1 links. Add another link to this network so that a loop is formed and eliminate from the loop the most costly link. Repeat until no further changes in the connecting network are possible. The resulting network T is the optimal network K. For suppose $T \neq K$ and that e_m is the link of smallest index in the Kruskal construction of Section 2 which is not in T. Add e_m to T to form a loop. This loop contains at least one link e_m' which is in T but is not in K. Furthermore adding e_m' to the set of Kruskal links $e_1, e_2, \ldots, e_{m-1}$ cannot complete a loop since all these links, including e_m', lie in the network T, which is free of loops. Therefore

(2) $$\text{price } (e_m) < \text{price } (e_m'),$$

so that the algorithm calls for adding e_m to T and eliminating e_m' from T, which is contrary to the assumption, under the rules of the algorithm, that no further changes in T are possible.

5. Solution V.

Though the algorithms mentioned above rather clearly show the structure of the minimizing network, they are not the best insofar as rapid computation of the solution is concerned. In this regard, a suggestion of Prim, [29], involving a combination of algorithms I and II, is probably best, for it avoids considerations of loops and connectedness, and makes rather modest memory requirements on a computing machine.

It is a simple matter to determine the most costly connecting network using similar procedures. Prim has also called attention to the fact that the minimizing connecting network K also minimizes all increasing symmetric functions and maximizes all decreasing symmetric functions of the link costs, among all connecting networks with no loops.

III. OPTIMAL PATHS THROUGH NETWORKS

In this section we shall discuss a variety of problems involving the determination of optimal paths through networks. The first of these, and perhaps the simplest, which will be attacked in several ways, involves the determination of a path of minimal time of transit between two points of a network. It is assumed that the time of transit of each link is known. It is then shown how the $n^{\underline{th}}$ shortest path (or paths) can be determined. The former problem is closely related to finding a path between two points in a network which has minimum probability of being blocked, given the probabilities that the individual links are blocked, and the fact of the independence of the individual links being blocked. Lastly an interesting extension is indicated which consists in determining a path between two points in a network which maximizes the probability of being traversed in a time t or less, being given the probability densities for the times of transit of the individual links and the information that the times of transit are independent.

It is clear that problems of the types just mentioned are of importance in the study of networks where the possibility of alternate routing exists. This will become even more apparent in Part IV in which it is shown that these problems are intimately connected with the general problem of optimal routing of messages in networks.

6. Formulation and Solution.

Given a network consisting of N stations and interconnecting links, where the time to traverse link (i,j) is $t_{ij} \geq 0$, $t_{jj} = 0$, find a shortest path from point 1 to point N. Note that t_{ij} need not equal t_{ji} and that t_{ij} need not be proportional to the distance between points i and j. Our first approach is based upon that given by Bellman, [2], in which the original problem is imbedded within the class of problems of determining the shortest paths from any point i in the network to the point N.

The problem is a combinatorial one in which we seek the minima of times of transit of a finite number of paths. For N of the order of twenty, the enumerative approach becomes quite onerous, so that we must determine more efficient methods of obtaining the extrema.

Denote the time of transit from i to N via an optimal path by u_i. Employing the principal of optimality, [1], we are led to the system of nonlinear equations

$$(1)\qquad \begin{cases} u_i = \underset{j \neq i}{\mathrm{Min}} \left\{ t_{ij} + u_j \right\}, \quad i = 1, 2, \ldots, N-1, \\ u_N = 0. \end{cases}$$

To resolve this system, following Bellman we resort to the method of successive approximations. As an initial approximation we set

$$(2)\qquad u_i^{(0)} = t_{iN}, \quad i = 1, 2, \ldots, N,$$

which corresponds physically to traversing the direct links from points i to point N. The higher approximations are then obtained through use of the formulas

$$(3)\qquad \begin{cases} u_i^{(k+1)} = \underset{j \neq i}{\mathrm{Min}} \left\{ t_{ij} + u_j^{(k)} \right\}, \quad i = 1, 2, \ldots, N-1, \\ u_N^{(k+1)} = 0, \end{cases}$$

for $k = 0, 1, 2, \ldots$. It is readily seen that $u_i^{(k)}$ is the minimal time of transit from point i to point N via k intermediate points. Since the sequence $u_i^{(k)} \geq 0$ is monotone non-increasing in k, the sequence converges to a solution of equation (1) in no more than N-2 iterations beyond the initial one. Furthermore, as Bellman has shown, the solution is unique, though an optimal path need not be.

This furnishes a feasible method for machine calculation with N of the order of several hundred. Since only additions and comparisons are required, the computation proceeds rapidly. Moreover, the memory requirement for the computation of $u_i^{(k+1)}$ is modest, since for each value of i only the $i^{\underline{th}}$ row of the matrix (t_{mn}) is required in addition to the previously computed values $u_j^{(k)}$.

It is also possible to obtain a monotone increasing sequence of approximations. Let

$$(4) \qquad \begin{cases} u_i^{(0)} = \underset{j \neq i}{\mathrm{Min}} \; t_{ij}, \quad i = 1, 2, \ldots, N-1, \\ \\ u_N^{(0)} = 0, \end{cases}$$

be the initial approximation, and let the additional approximations be determined by the relations in equation (3). We can see inductively that the sequence is monotone non-decreasing and furthermore that

$$(5) \qquad u_i^{(k)} \leq u_i, \; i = 1, 2, \ldots, N, \quad k = 0, 1, 2, \ldots,$$

where u_i is the solution of equation (1). For $k = 0$ the inequality (5) is valid. Hence if we assume it holds for $k = m$, we obtain

$$(6) \qquad u_i^{(m+1)} = \underset{j \neq i}{\mathrm{Min}} \left\{ t_{ij} + u_i^{(m)} \right\} \leq \underset{j \neq i}{\mathrm{Min}} \left\{ t_{ij} + u_i \right\} \leq u_i ,$$

which completes the induction and establishes the monotone convergence of the sequence defined by equations (3) and (4).

Observe that if a shortest path connecting 1 to N through one intermediate point, m, in required, the solution is given by the sum of the shortest chains connecting 1 to m and m to N. Should two intermediate points, m and n, be specified, the solution is the shorter of the chains (1,m,n,N) and (1,n,m,N) where each pair of nodes, (1,m), (m,n), (n,N), etc. is joined by a shortest chain. If the number of intermediate points is small, then a shortest path can be determined through enumeration of cases, the computation of a shortest path between two specified points being effected as above.

7. Solution II.

Another technique for solving the problem posed at the beginning of section 6 is contained in an algorithm described by Ford, [14], and others. It is, of course, simply another way of solving equations (6.1) and runs as follows. Assign the value $0 = u_N$ to the node N and $u_i = \infty$ to the nodes $i \neq N$. Hunt through the network until a pair of points i and j with the property that

$$(1) \qquad u_i > t_{ij} + u_j$$

is found, should there be any such. Then replace u_i at the node i with the smaller value $t_{ij} + u_j$. Repeat this step until no pairs fulfilling the inequality (1) remain. The numbers u_i then assigned to the nodes i represent the minimal times of transit from these nodes to the node N. This will now be proved.

Let i, i_1, i_2, ..., N be an optimal chain from i to N. We have

(2) $$u_i - u_{i_1} \leq t_{ii_1} ,$$

with similar inequalities holding for every link in the chain. Through addition of all these inequalities we find that

(3) $$u_i \leq \text{minimal time of transit from i to N.}$$

On the other hand, for every node m≠N there is a link from m to a node n for which

(4) $$u_m = t_{mn} + u_n.$$

All nodes except N were initially assigned the values ∞, and these values have been monotone decreasing (or else have not changed at all). At the last decrease in u_m there is an n which still has the same value. We can trace a chain from i to N composed of links for which equalities such as that in equation (4) hold. The values at the nodes are decreasing. Eventually the point N must be attained. Along this chain

(5) $$u_j - u_k = t_{jk}.$$

A summation yields that

(6) $$u_i = \text{time of transit of this chain.}$$

Consequently this is a shortest chain.

An elegant version of this algorithm has been suggested by Dantzig. It enables one to determine the minimal times of transit from 1 to N and the paths to be traversed through use of a constructive procedure reminiscent of Kruskal's algorithm. First determine a closest point to N, say P_1, and record the time of transit from P_1 to N. Then determine a closest point to P_1, say Q, and also a point which is second closest, via a direct path, from N, say R. Determine the smaller of t_{RN} and $t_{QP_1} + u_{P_1}$. This yields P_2, the second closest point to N, and an optimal path from P_2 to N. A comparison among the times to travel to N from the closest unchosen point to N, via a direct path, and the closest points to P_1 and P_2, continuing from P_1 or P_2 along the paths already selected, yields P_3, and so on.

If there are N stations in the network, this procedure will result in solution after at most $1 + 2 + (N-1) = \frac{N-1}{2} N$ comparisons. This assumes that for each node in the network the remaining ones have been arranged in order according to the times of transit from the given node to each of the others.

8. The n^{th} Shortest Chains.

It has been noted by Bellman that the n^{th} shortest paths can also be conveniently determined through use of functional equations. The importance of this resides in the fact that this enables us to see how sensitive to change the times of transit are for paths in neighborhoods of optimal paths. This has implications for the general theory of multi-stage decision processes which will be discussed elsewhere, [5].

We define u_i, $i = 1, 2, \ldots, N$, as in the previous section and introduce the quantities

(1) v_i = time of transit of a second shortest path from i to N,

for $i = 1, 2, \ldots, N-1$.

Next we observe that if the first link in a second shortest route is the link (i,j) then the continuation from j to N must be along either a path which minimizes the time of transit from j to N or which is a second shortest path from j to N, no others being possible. These lead to total durations of the routes from i to N of $t_{ij} + u_j$ and $t_{ij} + v_j$ respectively. Hence v_i is equal to the smaller of the following two values: the second smallest value of $t_{ij} + u_j$, $j \neq i$, and the smallest value of $t_{ij} + v_j$, $j \neq i$. If Min_k refers to the operation of taking the k^{th} smallest value of a given set, with $\text{Min}_1 = \text{Min}$, the resulting equations are

$$(2) \qquad v_i = \text{Min}\left\{ \underset{j \neq i}{\text{Min}_1} (t_{ij} + v_j), \underset{j \neq i}{\text{Min}_2} (t_{ij} + u_j) \right\} ,$$

for $i = 1, 2, \ldots, N-1$.

The generalization to the calculation of the n^{th} shortest paths is evident, though various questions concerning the numerical solution of the equations arise.

9. Solutions by Analogue Computation.

The problem of determining the shortest path between two points in a network may also be solved by constructing a string model, [34], in which inextensible strings of lengths proportional to the times of transit are connected between all pairs of nodes in a network. A path of minimal time of transit between two nodes is then determined by separating the selected pair of nodes to the greatest extent possible. The links in chains which are stretched taut form optimal paths, and the distance of separation of the points measures the time of transit over an optimal path.

Electrical analogues can also be employed. Each branch of the network is replaced by gas tubes whose breakdown voltage is proportional to the times of transit, and the terminals of a current source are connected to points under consideration. The paths over which current flows are optimal.

See also [26] for a discussion of related matters, including use of soap-film models.

10. Some Stochastic Problems.

We now turn our attention to some extensions in which various probabilistic elements are introduced. Consider a switching network in which the probability that a link from m to n is available for service is p_{mn}. The problem is to determine a path from 1 to N which has the greatest probability of being available for service (i.e., unblocked).

We introduce a set of variables P_i, $i = 1, 2, \ldots, N$, defined by the relation

$$\text{(1)} \qquad P_i = \text{the probability of no blocking on an optimal path from } i \text{ to the point } N.$$

This leads to the relations

$$\text{(2)} \qquad \begin{cases} P_i = \underset{j \neq i}{\text{Max}} \; p_{ij} \, P_j, \; i = 1, 2, \ldots, N-1, \\ P_N = 1, \end{cases}$$

which, similarly to the equations discussed earlier, can be resolved through use of the successive approximations

$$\text{(3)} \qquad \begin{cases} P_i^{(k+1)} = \underset{j \neq i}{\text{Max}} \; p_{ij} \, P_j^{(k)}, \; i = 1, 2, \ldots, N-1, \\ P_N^{(k+1)} = 1, \end{cases}$$

for $k = 0, 1, 2, \ldots$, along with the initial approximation

$$\text{(4)} \qquad \begin{cases} P_i^{(0)} = p_{iN}, \\ P_N^{(0)} = 1. \end{cases}$$

The sequence is clearly monotone increasing.

Now let us suppose that the time to traverse the link from i to j is a random variable t_{ij} with probability density function $p_{ij}(s)$, $i \neq j$, $s \geq 0$, and that the times of transit of the various links are independent. The treatment of the problem in which we seek a path from i to N for which the average time of transit is minimum is evident. Let us therefore turn to the problem in which we require a path connecting the point i to the point N which maximizes the probability that the time of transit is no greater than a given time t. Again using the principal of optimality, after introducing the functions $u_i(t)$, $i = 1, 2, \ldots, N$, to be the probability that the time of transit from i to N is no greater than t, using an optimal path, we find

$$(5)\quad \begin{cases} u_i(t) = \max\limits_{j \neq i} \displaystyle\int_0^t p_{ij}(t - s)\, u_j(s)ds, & i = 1, 2, \ldots, N-1, \\ u_N(t) = 1. \end{cases}$$

Once again we may resort to the method of successive approximations to resolve this nonlinear system:

$$(6)\quad \begin{cases} u_i^{(k+1)}(t) = \max\limits_{j \neq i} \displaystyle\int_0^t p_{ij}(t - s)\, u_j^{(k)}(s)ds, & k = 0, 1, 2, \ldots \\ u_N^{(k+1)}(t) = 1. \end{cases}$$

As initial approximations we take

$$(7)\quad \begin{aligned} u_i^{(0)}(t) &= \int_0^t p_{iN}(s)ds, \quad i = 1, 2, \ldots, N-1, \\ u_N^{(0)}(t) &= 1, \end{aligned}$$

which yields approximations that are monotone increasing. The initial approximation

$$
\begin{aligned}
u_i^{(0)}(t) &= \operatorname*{Max}_{j \neq i} \int_0^t p_{ij}(s)\,ds, \quad i = 1, 2, \ldots, N-1, \\
u_N^{(0)}(t) &= 1,
\end{aligned}
\tag{8}
$$

yields monotone decreasing approximations.

IV. OPTIMAL ROUTING PROBLEMS

A problem of considerable importance in the operation of communication systems is that of the determination of the routing doctrine to be used in handling the messages. Large systems frequently employ a central traffic control unit for this purpose. Information concerning backlogs of messages are periodically sent to this control unit, as is information concerning the state of the communication system itself (wires may be down, equipment may be malfunctioning, etc.). On the basis of this information plus predictions concerning the new demands for service, decisions are made concerning the way the messages are to be routed through the network. Inefficiencies in the routing of the messages are reflected in the need for greater quantities of equipment for a fixed grade of service.

The papers referred to in the introduction, some of which contain extensive bibliographies, indicate a mathematical treatment of these problems based on probability theory. Interest centers on fluctuations in the traffic. Here we shall consider a steady state formulation for these problems which leads to a linear programming setting. A further discussion can be found in [20].

Even for moderately sized networks of about thirty stations the problems become so large that solution is not feasible through use of the general simplex method of George Dantzig, [11]. Instead we resort to use of a modification of the simplex method which was originally proposed for multi-commodity flow problems and which is due to Ford and Fulkerson, [16]. First the general approach is sketched and then a simple optimal routing problem is worked in detail for illustrative purposes.

11. Problem Formulation and Method of Solution.

We now reduce this version of the problem of the routing of messages in a network to mathematical form. Introduce the quantities

(1) d_{ij} = the number of messages available at i which are destined for j,

(2) c_{ij} = the number of messages which can be sent over the direct link from i to j.

All action is assumed to take place during a given time interval. Next label all the directed links in the network $L_1, L_2, \ldots, L_m$, and label all the directed routes in the network which lead from a source to a destination $R_1, R_2, \ldots, R_n$. We describe the composition of the routes in terms of the links through use of the $m \times n$ incidence matrix (a_{ij}), where

$$(3) \qquad a_{ij} = \begin{cases} 1, & \text{if link } i \text{ lies in route } j \\ 0, & \text{otherwise.} \end{cases}$$

If the link from i to j is labelled s, we set

$$(4) \qquad c_{ij} = c_s.$$

At each source S_i it is convenient to modify the original network by introducing a set of fictitious sources, $S_i^{(r)}$, which are connected to S_i by fictitious directed links, each fictitious source $S_i^{(r)}$ corresponding to messages originated at i which are destined for the station r. The capacity of each fictitious link $(S_i^{(r)}, i)$ is d_{ir}. If d_{ir} is zero, then the fictitious source and link are not introduced. In this way all messages are conceived of as arising at fictitious sources; the messages flow over the fictitious links and then the actual links to their destinations.

Hence all constraints, as in relations (6) and (7) below, appear as capacity constraints, including those that are due to the limited supplies of messages available for delivery. All routes lead from fictitious sources to their destinations, and we shall assume that the incidence matrix (a_{ij}) has reference to the modified network. In particular m is the sum of the numbers of actual and fictitious links. We shall henceforth not distinguish between fictitious stations and actual stations. Lastly we let x_j be the number of messages which flow over the route j, j = 1, 2,..., n.

The problem involves the maximization of the number of delivered messages

$$(5) \qquad d = \sum_{j=1}^{n} x_j ,$$

subject to the constraints

$$(6) \qquad x_j \geq 0, \; j = 1, 2, \ldots, n + m,$$

$$(7) \qquad \sum_{j=1}^{n} a_{sj} x_j + x_{n+s} = c_s .$$

Here we have denoted the amount of unused capacity in link s by x_{n+s}, s = 1, 2, ..., m.

If the problem is to maximize the revenue derived from the operation and r_j is the return from sending a message over the $j^{\underline{th}}$ route, then the objective form becomes

$$(5') \qquad r = \sum_{j=1}^{n} r_j x_j .$$

As was remarked earlier, n, the number of routes becomes so large, even in moderately sized networks, that it is not possible to determine an optimal linear program through use of the simplex method in its most general form, even if use of a high-speed digital computer is contemplated. The memory requirements for storage of the matrix (a_{ij}) alone become excessive. Hence we resort to a modification of the simplex method in which only m columns of the matrix, the basic vectors, need be stored simultaneously. At each stage of the simplex algorithm the new vector to be brought into the basis is determined by several applications of one of the algorithms described earlier for determining a shortest path connecting two points in a network.

From the general theory of linear inequalities we know that there is an optimal routing of the messages in which no more than m of the activities of sending a message over a route or storing capacity on a link are raised above the zero-level. Using this fact we can place the entire algorithm on quite an intuitive basis. We start by storing all capacity on all links, so that $x_{n+s} = c_s$, $s = 1, 2 \ldots, m$. We then show how to improve the routing doctrine at any stage of the process by raising some favorable activity from zero-level to some positive level which is high enough to drive the level of some other formerly nonzero-level activity down to zero-level. To determine which new activity to introduce we consider the 'shadow' prices which are induced on the capacities as a result of non-zero activities which are carried out at a particular stage. See [12] for a general discussion. The prices that are assigned to the fictitious links may be thought of as being franchise prices, that is, unit prices of the right to accept messages at one station destined for another. Prices assigned to the actual links are unit prices for the equipment.

Let p_j be the value of each unit of capacity in link j. For each additional send ng of a message over a route j, j = 1, 2, ..., n, the number of messages delivered is increased by unity. The unit prices of the capacities in the links along the routes used must therefore sum to unity,

$$(8) \qquad \sum_{s=1}^{m} a_{sj}\, p_s = 1, \text{ j such that } x_j > 0, \; j \leq n.$$

On the other hand there is a zero return for storing capacity so that

$$(9) \qquad p_{j-n} = 0, \; x_j > 0, \; j > n.$$

The equations (8) and (9) determine the m prices $p_1, p_2, \ldots \; p_m$.

Let us now introduce an entrepreneur who examines the system capacities, the users' demands and the price structure in an effort to determine whether or not it is possible to buy capacity from the communication network operating company, according to the price schedule $\{p_i\}$ and deliver messages himself at a profit, a delivered message being worth one unit. That is, the entrepreneur wishes to ascertain whether or not there is a route j for which

$$(10) \qquad \sum_{s=1}^{m} a_{sj}\, p_s < 1, \quad j = 1, 2, \ldots, n.$$

If there is such a route, though, it would be advantageous for the operating company to send messages over that route to the greatest extent possible. In general sending messages over route j will use capacity that was being used for sending other messages, so that some other activities may have to be curtailed, until finally at least one is reduced to zero-level, and so is eliminated. In any event capacity constraints prevent x_j from increasing indefinitely.

Should the price of a certain link be negative, then this may be interpreted to mean that the communication system operating concern would be willing to pay the entrepreneur a subsidy to take this capacity from it. Rather than do this, this capacity should be sent to storage, so that if p_j is negative it is advantageous to raise x_{n+j} above the zero-level.

Assuming that all the prices are positive, how can the entrepreneur determine a route for which condition (10) is fulfilled? Since each unit of capacity is assigned a price, including the capacity of the fictitious links, he has merely to determine a lowest-price route from each source to destination. As soon as one is found for which the price is less than unity, as many messages as possible should be sent from this source to destination. If there is no such route, then the routing doctrine being employed is optimal, as one sees from the duality theorem of linear programming. This idea constitutes the essence of the delightfully simple suggestion of Ford and Fulkerson.

To summarize, the steps in the algorithm are

1. Under the current price schedule determine a favorable activity to introduce. If a price is negative, store as much as possible of the corresponding capacity; otherwise determine, via one of the algorithms discussed in Part III, a route having cost

less than unity, and introduce this activity. If there is none, the routing doctrine is optimal.

2. Increase the level of the favorable activity until some activity which was previously at a nonzero-level is driven to zero-level. This determines the new routing doctrine and the number of messages which are thereby delivered.

3. Determine the new schedule of unit prices on the capacities and return to Step 1.

An illustrative example is provided below to illustrate this technique.

If the total number of links, including the fictitious ones, is of the order of 150, the steps of the algorithm are possible for implementation on a high-speed computing machine. It is difficult to try to estimate the rate at which the approximations coverage to an optional solution since the number of chains might be numbered in the tens of thousands. Some numerical experimentation is undoubtedly called for. In actual computations great advantages might be realized by being very selective with regard to which favorable activity is to be introduced at each stage.

12. Solution of an Illustrative Optional Routing Problem.

Consider the four-station network shown below in Fig. 1 in which the capacities of the links are as shown. We assume that the link capacities are undirected rather than directed, a matter of no importance insofar as the method is concerned. Consider that station 4 has 5 messages destined for station 2 and station 1 has [illegible]

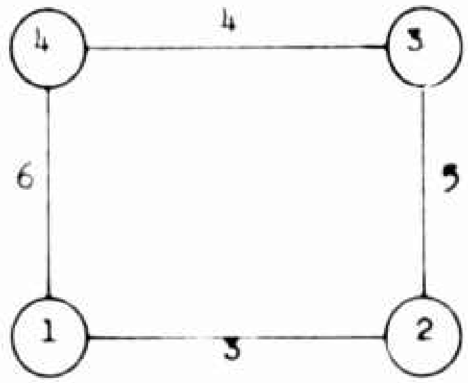


Fig. 1 - A Capacitated Four-Station Network

messages destined for station 3. This is accounted for in Fig. 2 in which the appropriate artificial stations and links are introduced. Station B has messages destined for 2 and station A has messages destined for 3. Since there are six links. there is an optional solution with no more than six activities raised above the zero-level.

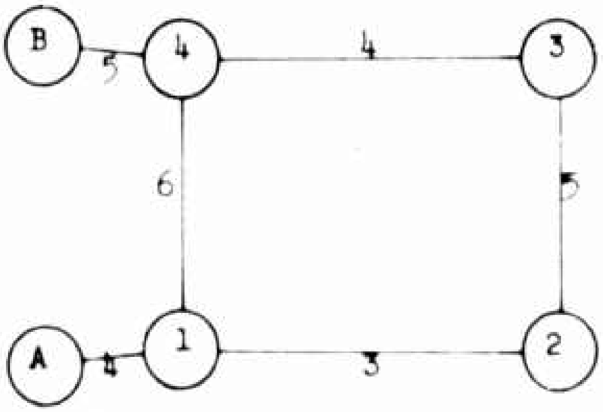
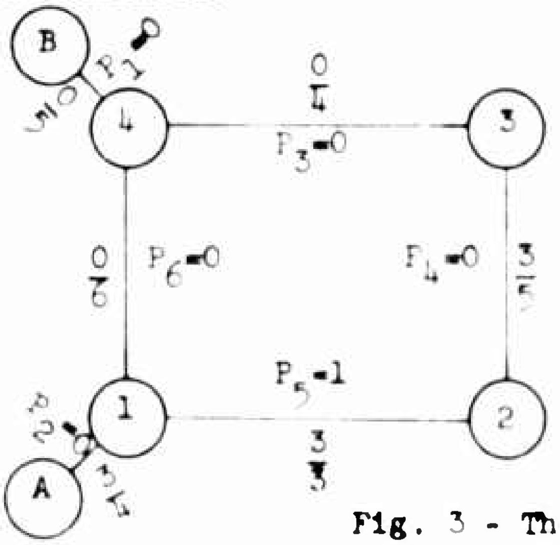


Fig. 2 - The Network Including the Artificial Elements

To start the algorithm we put all capacity in storage, which corresponds to backlogging all messages. Since no messages are delivered, all capacities have prices of zero. By inspection, we see that the unit cost for the route (A, 1, 2. 3) is zero; consequently three messages are sent over this route. This eliminates the activity of storing capacity in link (1, 2). Letting the unit price of capacity in link (B, 4) be p_1, that in (A, 1) be p_2. and so on, as shown in Fig. 3, we find that the prices satisfy the equations

$$\begin{aligned} &p_1 = p_2 = p_3 = p_4 = p_6 = 0, \\ &p_2 + p_5 + p_4 = 1. \end{aligned} \tag{1}$$

The situation is shown in Fig. 3 in which the amounts of capacity used are shown above the horizontal lines on the links and the capacities below them. The prices implied by equations (1) are also indicated.



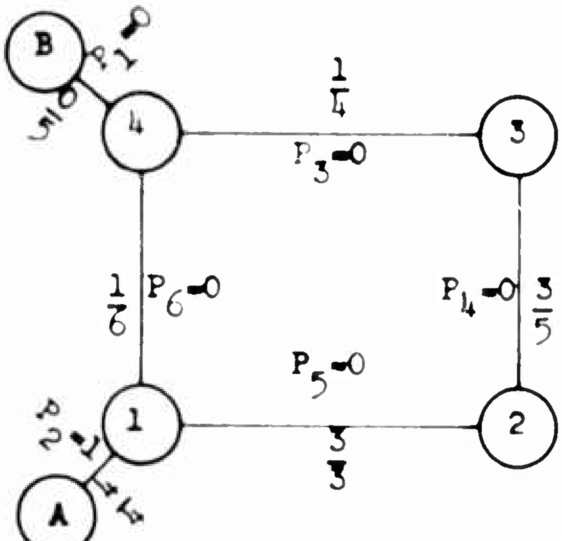
No. of Msgs.	Route
3	(A, 1, 2, 3)

3 msgs. delivered

Fig. 3 - The First Stage

The unit price of the route (A, 1, 4, 3) is zero. One message is sent over this route, which eliminates the activity of storing capacity along the fictitious link (A, 1) (i.e., the activity of backlogging messages at A is eliminated). With this routing schedule the equations for the new prices becomes

$$\begin{aligned} &p_1 = p_3 = p_4 = p_6 = 0, \\ &p_2 + p_5 + p_4 = 1, \\ &p_2 + p_6 + p_3 = 1. \end{aligned} \tag{2}$$



No. of Msgs.	Route
3	(A, 1, 2, 3)
1	(A 1, 4, 3)

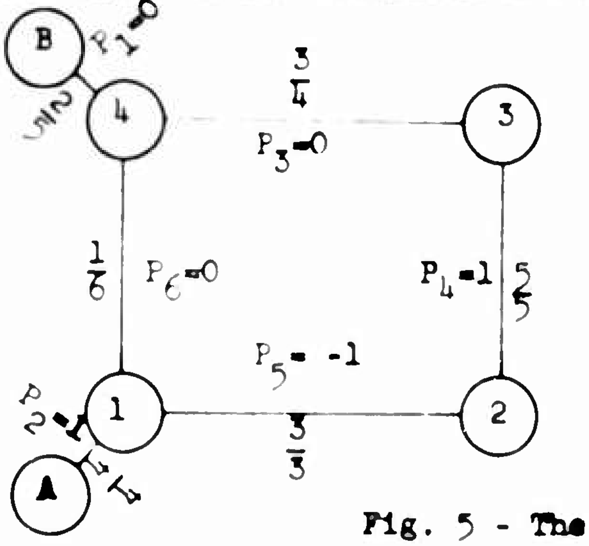
4 msgs. delivered

Fig. 4 - The Second Stage

The unit price of the route (B, 4, 3, 2) is zero. Two messages are sent over this route which saturates link (3, 2). The new prices are determined from the equations

$$
(3)\qquad
\begin{aligned}
&p_1 = p_3 = p_6 = 0,\\
&p_2 + p_5 + p_4 = 1,\\
&p_2 + p_6 + p_3 = 1,\\
&p_1 + p_3 + p_4 = 1,
\end{aligned}
$$

which leads to the situation of Fig. 5.

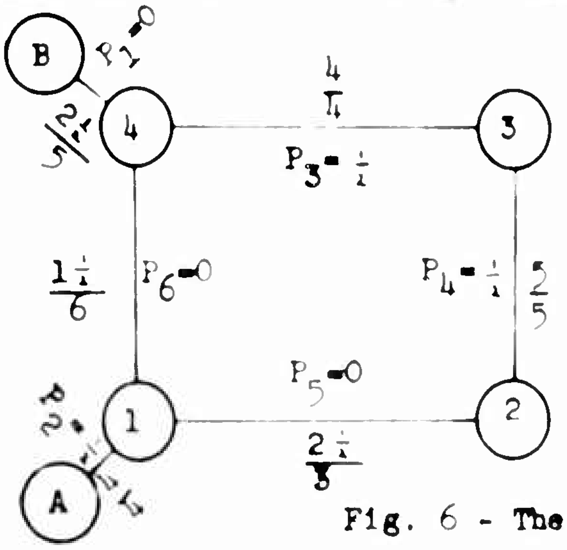


No. of Msgs.	Route
3	(A, 1, 2, 3)
1	(A, 1, 4, 3)
2	(B, 4, 3, 2)

6 msgs. delivered

Fig. 5 - The Third Stage

Since the unit price p_5 is negative, the activity of storing capacity on the link (1, 2) is reintroduced in the amount z. To find which activity is eliminated we note that 3-z messages are then sent over the route (A, 1, 2, 3), which causes 2 + z to be sent over (B, 4, 3, 2), to avoid introducing the storage of capacity on the link (2, 3). The number of messages sent over (A, 1, 4, 3,) must be increased to 1 + z to avoid introducing the backlogging of messages at station A. Then an examination of the flows over each link shows that z can be increased to 1/2, at which point the link (4, 3) is saturated, so that storage of capacity on this link is eliminated. The new routing and price schedule are shown in Fig. 6.



No. of Msgs.	Route
2 1/2	(A, 1, 2, 3)
1 1/2	(A, 1, 4, 3)
2 1/2	(B, 4, 3, 2)

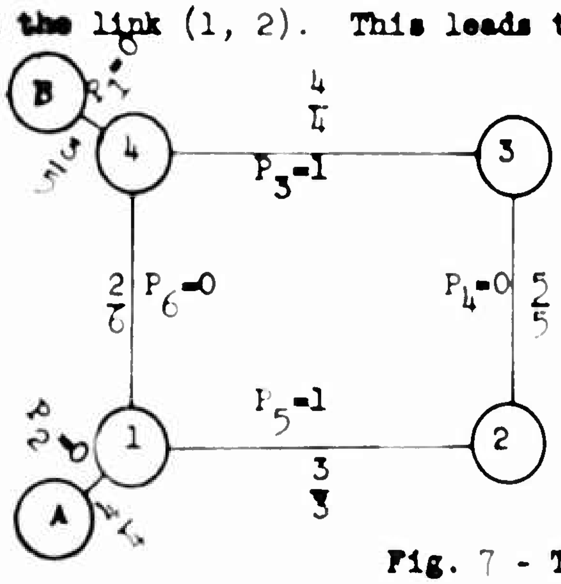
6 1/2 msgs. delivered

Fig. 6 - The Fourth Stage

The prices are determined from the equations

$$
(4) \quad
\begin{aligned}
p_1 &= p_5 = p_6 = 0, \\
p_2 + p_5 + p_4 &= 1, \\
p_2 + p_6 + p_3 &= 1, \\
p_1 + p_3 + p_4 &= 1.
\end{aligned}
$$

The route (B, 4, 1. 2) now has zero unit price, so that 1/2 message is sent along this route, which eliminates storage of capacity on the link (1, 2). This leads to the situation of Fig. 7.



No. of Msgs.	Route
2 1/2	(A, 1, 2, 3)
1 1/2	(A, 1, 4. 3)
2 1/2	(B, 4, 3, 2)
1/2	(B, 4, 1, 2)

7 msgs. delivered

Fig. 7 - The Last Stage

The prices are determined from the equations

$$p_1 = p_6 = 0,$$

$$p_2 + p_5 + p_4 = 1$$

(5) $$p_2 + p_6 + p_3 = 1,$$

$$p_1 + p_3 + p_4 = 1,$$

$$p_1 + p_6 + p_3 = 1.$$

The solution is optimal, for, as is seen from the figure, no prices are negative and no paths from origin to destination exist which have unit costs of less than one.

By the way of comment it should be pointed out that in solution of large scale problems much more efficient methods of computing the prices and determining the new activity levels, as we proceed from stage to stage, are available. The method adopted here is for illustrative purposes only.

The unit prices shown in Fig. 7 show where the real bottlenecks in the system are. Thus for z sufficiently small, if the capacity of link (1, 2) is increased by the amount z, then z additional messages can be sent over the route (B, 4, 1, 2). The same holds true for the link (4, 3), though this is not so obvious. This rests on the observation that the solution is not unique. If messages are recalled along each route from B to 2, as is clearly possible, since the flow on each link is both increased and decreased by x, then for $2 x = x$ sufficiently small, the messages which are then backlogged at A could be sent via the route (A, 1, 4, 3) to their destination.

V. DISCUSSION

The problems mentioned earlier should be viewed merely as suggestive of a host of other essentially combinatorial problems which arise in the general field of communication. We shall now single out a few for further discussion; still others can be found by checking the list of references provided, not all of which are referred to in the text.

A routing problem which has been studied extensively is the one which requires the determination of the maximum steady state flow of a homogeneous commodity through a capacitated network from a source to a sink. Boldyreff's flooding technique is described in, [6], and the minimum cut maximum flow theorem is proved in, [15], where additional references can be found.

It is apparent that many combinatorial problems arise in the design and utilization of switching networks, [22, 19]. In the absence of suitable analytic techniques for handling such problems, one frequently resorts to the use of simulation devices known as 'throwdown' machines, [17]. With reference to blocking in networks, [9, 25], it would be of interest to find general and efficient techniques for calculating the probability of finding at least one path available from a given point 1 to another point N in a network, under various assumptions concerning the probabilities of finding the individual links available. The functional equation technique of Part III does not appear to be immediately applicable, as in the determination of a path with highest probability of being available.

Interesting treatments of the effects of congestion in the networks, from still a different viewpoint, can be found in Wardrop, [32], Prager, [28], and Charnes and Cooper, [8].

The solution of the optimal routing problem discussed in Part IV can be directly applied to the determination of optimal interoffice trunking arrangements as is indicated in [21]. Other methods of solution, based on the primal-dual algorithm, are discussed in W. Jewell's M.I.T. doctoral dissertation (1958).

Lastly mention may be made of the problem of determining minimal cost augmentations to be made to a given system to provide a satisfactory grade of service in view of anticipated increases in future demands for service. These are given a linear programming formulation in [20]. Much work remains to be done, however, in order to find efficient methods of solution.

REFERENCES

1. R. Bellman, *Dynamic Programming*, Prin. Univ. Press, Princeton, 1957.

2. R. Bellman, 'On a Routing Problem,' *Quart. Of Appl. Math.*, V. 16 (1958), pp. 87-90.

3. R. Bellman, 'Notes on the Theory of Dynamic Programming -- Transportation Models,' *Manag. Sci.*, V. 4 (1958), pp. 191-195.

4. R. Bellman, 'Combinatorial Processes and Dynamic Programming ' these Proceedings.

5. R. Bellman and R. Kalaba, 'On k^{th} Best Policies,' to appear.

6. A. Boldyreff, 'Determination of the Maximal Steady State Flow of Traffic Through a Railroad Network,' *Oper. Res.*, V. 3 (1955), pp. 443-466.

7. E. Brockmeyer, H. Halstrøm, and A. Jensen, *The Life and Works of A. K. Erlang*, Copenhagen, 1948.

8. A. Charnes and W. Cooper, 'Extremal Principles for Simulating Traffic Flow in a Network,' *Proc. Nat. Acad. Sci. USA*, V. 44 (1958), pp. 201-204.

9. C. Clos, 'A Study of Non-Blocking Switching Networks,' *Bell Syst. Tech. J.*, V. 32 (1953), pp. 406-424.

10. G. B. Dantzig, 'Discrete-Variable Extremum Problems,' *Oper. Res.*, V. 5 (1957), pp. 266-277.

11. G. B. Dantzig, A. Orden and P. Wolfe, 'The Generalized Simplex Method for Minimizing a Linear Form under Linear Inequality Restraints,' *Pacific J. of Math.*, V. 5 (1955), pp. 183-195.

12. R. Dorfman, P. Samuelson and R. Solow, *Linear Programming and Economic Analysis*, McGraw-Hill Book Company, Inc., New York, 1958.

13. P. Elias, A. Feinstein and C. Shannon, 'A Note on the Maximum Flow Through a Network,' IRE *Transactions on Information Theory*, V. IT-2 (1956), pp. 117-119.

14. L. R. Ford, Jr., 'Network Flow Theory,' The RAND Corporation Paper P-923, 1956.

15. L. Ford, Jr. and D. Fulkerson, 'A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem,' *Canadian J. of Math.*, V. 9 (1957) pp. 210-218.

16. L. Ford, Jr. and D. Fulkerson, 'A Suggested Computation for Maximal Multi-Commodity Network Flows,' The RAND Corporation Paper P-1114, 1958.

17. G. Frost W. Keister and A. Ritchie, 'A Throwdown Machine for Telephone Traffic Studies,' *Bell Syst. Tech. J.*, V. 32 (1953) pp. 292-359.

18. T. C. Fry. *Probability and Its Engineering Uses*, D. Van Nostrand Company. Inc., New York, 1928.

19. F. Hohn, 'Some Mathematical Aspects of Switching.' *Amer. Math. Monthly*, V. 62 (1955), pp. 75-90.

20. R. Kalaba and M. Juncosa, 'Optimal Design and Utilization of Communication Networks,' *Manag. Sci.*, V. 3 (1956) pp. 33-44.

21. R. Kalaba and M. Juncosa, 'Optimal Utilization and Extension of Interoffice Trunking Facilities,' *Comm. and Elect.*, Jan. 1959, pp. 998-1003.

22. W. Keister, A. Ritchie and S. Washburn, *The Design of Switching Networks*, D. Van Nostrand Company, Inc., New York. 1951.

23. D. König, *Theorie der Endlichen und Unendlichen Graphen*, reprinted by Chelsea Publishing Company, New York, 1950.

24. J. B. Kruskal, Jr., 'On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem,' *Proc. Amer. Math. Soc.*, V. 7 (1956), pp. 48-50.

25. C. Y. Lee, 'Analysis of Switching Networks,' *Bell Syst. Tech. J.*, V. 34 (1955), pp. 1287-1315.

26. W. Miehle, 'Link-length Minimization in Networks,' *Oper. Res.*, V. 6 (1958), pp. 232-243.

27. E. C. Molina, 'Application of the Theory of Probability to Telephone Trunking Problems,' *Bell Syst. Tech. J.*, V. 6 (1927), pp. 461-494.

28. W. Prager, 'Problems of Traffic and Transportation,' *Proc., Symposium on Operations Research in Business and Industry*, Kansas City, 1954, pp. 105-113.

29. R. C. Prim. 'Shortest Connection Networks and Some Generalizations.' *Bell Syst. Tech. J.*, V. 36 (1957), pp. 1389-1401.

30. C. E. Shannon, 'A Mathematical Theory of Communication.' *Bell Syst. Tech. J.*, V. 27 (1948), pp. 379-423 and pp. 623-656.

31. C. Truitt. 'Traffic Engineering Techniques for Determining Trunk Requirements in Alternate Routing Trunk Networks,' *Bell Syst. Tech. J.*, V. 33 (1954), pp. 277-302.

32. J. G. Wardrop, 'Some Theoretical Aspects of Road Traffic Research.' Proc. Inst. Civ. Engineers (London), V. 1 (1952), pp. 325-378.

33. R. I. Wilkinson, 'Theories for Toll Traffic Engineering in the U.S.A.,' Bell Syst. Tech. J., V. 35 (1956), pp. 421-514.

Additional reference added in proof:

34. G. J. Minty, "A Comment on the Shortest Route Problem", Oper. Res., v. 5 (1957), p. 724.